Interference Mitigation by Statistical Interference Modeling in an Impulse Radio UWB Receiver

Manuel Flury and Jean-Yves Le Boudec
EPFL, CH-1015 Lausanne, Switzerland, {manuel.flury, jean-yves.leboudec}@epfl.ch

***Abstract*— Some impulse radio UWB (IR-UWB) networks may allow concurrent transmissions without power control (for example MAC protocols that do not use power control, or co-existing, non-coordinated piconets). In such cases, it has been proposed to mitigate multi-user interference (MUI) at the physical layer, but existing proposals for interference mitigation do not account for the multipath nature of UWB channels. We address this problem and propose a receiver that employs a combination of statistical interference modeling and thresholding to mitigate MUI. We find that in a multipath environment the proposed receiver significantly outperforms existing receiver designs that either completely neglect the effect of MUI or only use a simple threshold to reject samples from interfering users. Further, in contrast to successive interference cancellation schemes, our receiver does not require active decoding of each interferer. Thus there is no need to synchronize the receiver with all the interfering users, which would be impractical in an IR-UWB system that is likely to be run in ad hoc mode. To model MUI we consider a hidden Markov model (HMM) and a Gaussian mixture model (GMM). We find that the HMM models interference better than the GMM. However, the resulting performance difference is not huge and comes at the cost of increased receiver complexity.**

I. Introduction

Some impulse radio UWB (IR-UWB) networks may allow concurrent transmissions without power control (for example MAC protocols that do not use power control [1], or co-existing, non-coordinated piconets). In such systems multi-user interference (MUI) can be significant and have a large impact on the performance. Still, most of the current receiver proposals seem to neglect the effect of interference in the receiver design completely, or focus on interference from other (narrowband) systems. This is often motivated by the infrequent, low duty cycle transmissions of IR-UWB systems. However there are a lot of imaginable applications where specific events trigger a simultaneous transmission from a large number of nodes (e.g. detection of a fire outbreak). To guarantee the flawless functioning of any UWB-based application in these crucial situations, MUI has to be taken into account already in the design phase. Further, the severe multipath found in indoor environments increases the probability of pulse collisions despite the short time duration of UWB pulses.

It has been shown that MUI in an IR-UWB system is not accurately modeled through a Gaussian approximation [2], [3]. Receivers that do not mitigate MUI assume that it is part of the AWGN background noise and are thus likely to suffer from a large performance loss. In systems that do not use power control, some form of interference mitigation at the physical layer is thus needed. Consequently, a threshold based approach has been suggested in [1]. The receiver defines a threshold based on the estimated power of the signal of interest and the estimated level of the background noise. It then declares samples that lie above the threshold as erasures, as they are likely to have a high interference term. However, this proposal does not account for the multipath nature of UWB channels. In a near-far scenario, even highly attenuated paths cause significant interference. This makes it very difficult to accurately estimate the signal and noise power and to determine where to actually set the threshold. On the other hand, classical solutions like successive interference cancellation require active decoding of each interferer. Therefore they are not a good solution for an IR-UWB system that is likely to be run in ad hoc mode as active decoding makes it necessary to synchronize the receiver with all the interfering users.

We address these problems and propose a receiver that employs a combination of statistical interference modeling and thresholding to mitigate MUI. As already mentioned, a Gaussian model is not well suited for MUI in an IR-UWB system. A popular non-Gaussian model for MUI is the Gaussian Mixture Model (GMM), see e.g. [4]. The GMM assumes that the interference has an underlying probability distribution formed by a mixture of Gaussians with different variances. Each interference term is then assumed to be generated by one of these mixture components. The GMM thus seeks to classify each sample and typically attributes samples with high interference to mixture components with high variances. In [5], the GMM has been proposed as MUI model for IR-UWB and it has been shown how to do channel and interference statistics estimation based on this model. Our paper takes the approach taken in [5] one step further and shows how to use these estimates to mitigate interference in the decoding phase. We also show that mitigation through interference modeling alone is not sufficient and that some kind of thresholding is still needed. Further, we also wanted to explore how much we can gain by considering a more sophisticated MUI model than the GMM proposed in [5]. The GMM assumes that the mixture components are independently chosen. However, due to the multipath nature of the channel this is not necessarily true as samples with a high interference level are likely to occur in bursts. Therefore, we propose to introduce correlation by modeling the sequence of mixture components with a homogeneous Markov chain. The resulting MUI model is a hidden Markov model (HMM) where each state is associated

The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

with a Gaussian output distribution. The GMM is just a special case of the more general HMM where the choice of the next state is independent of the current state. Obviously the HMM introduces a higher degree of complexity. Note that we are for the moment only interested in a possible gain from a theoretical viewpoint and are thus not considering how to actually implement it in a low-complexity real-world system.

Our main contributions are the following: (1) We propose a coherent RAKE receiver that takes MUI into account and therefore effectively mitigates the effect of MUI in a multipath environment (the algorithm is given in the end of section III-D). (2) We identify the need to classify MUI into three different types and to deal with each of them in an appropriate manner (section III-A). (3) We propose a new and more general model for MUI in IR-UWB based on a HMM (sections I and III-B). (4) We make a performance evaluation comparing different methods and models and assessing their ability to mitigate MUI in a realistic multipath environment (section IV).

II. System Model and Assumptions

We are studying an impulse radio physical layer with time-hopping and binary phase shift keying (BPSK) modulation. We assume that physical layer transmissions occur in packets of N_d data symbols. The signal emitted by the ith transmitter is then

$$s^{(i)}(t) = \sum_{j=-\infty}^{\infty} \sum_{k=1}^{N_d} d_{kj}^{(i)} \cdot p(t - c_k^{(i)} T_c - kT_f - \tau_{p,j}^{(i)}) \quad (1)$$

where $d_{kj}^{(i)} \in \{\pm 1\}$ is the kth symbol of the jth packet of the BPSK modulated data sequence of user i, $p(t)$ is the pulse shape of duration T_p, T_c is the length of a chip (we assume $T_c \geq T_p$), T_f is the frame duration, $\tau_{p,j}^{(i)}$ denotes the time the transmission of the jth packet of user i starts, and finally $c_k^{(i)}$ is the time-hopping sequence (THS) of user i. We assume that each user has its own pseudo-random THS that is known at the receiver. We further require the THS to be constrained such that no inter-symbol interference (ISI) occurs. The THS is integer valued and uniformly distributed on $[0, N_c - 1]$, where N_c is the number of chips available for time-hopping. The length of a frame equals $T_f = N_c \cdot T_c + T_g = (N_c + N_g) \cdot T_c$ where N_g is the number of guard slots preventing ISI and T_g is the length of the guard interval.

The multipath channel between user i and the receiver is characterized by the impulse response $h^{(i)}(t) = \sum_{l=0}^{L-1} \alpha_l^{(i)} \delta(t - \tau_l^{(i)})$, where L is the maximum number of paths, $\alpha_l^{(i)}$ is the channel coefficient of the lth path, δ is the dirac function and $\tau_l^{(i)}$ is the delay of the lth path. We assume that the channel is invariant for the duration of one packet.

The signal at the receiver is $r(t) = \sum_{i=1}^{N_u} s^{(i)}(t) * h^{(i)}(t) + n(t)$, where N_u is the number of users present in the system, $*$ denotes convolution and $n(t)$ is AWGN with zero mean and double sided power spectral density $\frac{N_0}{2}$. Assume that the user of interest is $i = 1$. For simplicity, we only consider the reception of one packet which allows us to drop the sum over j for $i = 1$ in (1). We further assume that synchronization has been achieved by some means, therefore $\tau_{p,j}^{(1)} = 0$. The received signal is then

$$r(t) = \sum_{l=0}^{L-1} \alpha_l \sum_{k=1}^{N_d} d_k \cdot p(t - c_k T_c - kT_f - \tau_l) + v(t) \quad (2)$$

where $v(t) = \sum_{i=2}^{N_u} s^{(i)}(t) * h^{(i)}(t) + n(t)$ accounts for MUI and background noise and where we dropped the index i for the user of interest to ease notation. The receiver performs matched filtering with a filter matched to the transmit pulse $p(t)$ and samples the filtered signal at the chip level to obtain the discrete time signal

$$y_n = \sum_{l=0}^{L-1} \alpha_l \sum_{k=1}^{N_d} d_k \cdot \mathcal{R}_p(nT_c - c_k T_c - kT_f - \tau_l) + z_n \quad (3)$$

where $\mathcal{R}_p(\tau)$ is the autocorrelation function of $p(t)$, $z_n = \sum_{i=2}^{N_u} s^{(i)}(t) * h^{(i)}(t) * p(-t) + w_n$ with $w_n \sim \mathcal{N}(0, \frac{N_0}{2} R_p(0))$. Assuming that the guard interval is properly designed to prevent ISI, (3) can be written as

$$y_n = \sum_{k=1}^{N_d} d_k \sum_{m=0}^{M-1} \beta_m \cdot \delta(n - c_k - k(N_c + N_g) - m) + z_n \quad (4)$$

where $\beta_m = \sum_{l=0}^{L-1} \alpha_l \mathcal{R}_p(mT_c - \tau_l)$ and $M = \left\lceil \frac{T_{\mathcal{R}_p} + T_{ch}}{T_c} \right\rceil$ with $T_{\mathcal{R}_p}$ denoting the duration of $\mathcal{R}_p$ and T_{ch} the channel spread[1]. Equation (4) is the discrete time representation of our system and will be used throughout the rest of this paper.

III. Interference Mitigation

We are now going to introduce our solution to interference mitigation at the receiver. We are following a data-aided approach, meaning that part of the data sequence, the training sequence, is known to the receiver. The N_d data symbols of a physical layer packet are thus divided into two parts: The first N_t symbols constitute the above mentioned training sequence (which is typically short in comparison to the length of the packet), the remaining N_p samples contain the information to be transmitted, the payload[2]. Accordingly, our receiver proceeds in two phases: A training phase and a data reception phase. In the training phase, the channel coefficients β_m as well as the statistics of z_n are estimated based on the known training sequence. In the data reception phase, these estimates are then used to mitigate the effect of interference and to recover the unknown data sequence.

A. Taxonomy of interference types

Before going into details of our receiver design, we analyze the different interference scenarios we are facing. This gives us a better understanding of what can happen and where the challenges are.

If interference occurs during packet reception, it must fall into one of the following three categories:

1) Interference is present during both training and data reception (called interference of type 1 from here on)
2) Interference is present during training only (type 2)
3) Interference present during data reception only (type 3)

[1]We can think of $h_n = \sum_{m=0}^{M-1} \beta_m \cdot \delta(n - m)$ as an equivalent discrete time channel impulse response.

[2]In a complete system, there actually is a third part preceding the training sequence. This part, the synchronization preamble, is used for signal acquisition and synchronization. As we assume that synchronization has already been obtained, we drop this part here for simplicity. However we still account for it in our simulations, see section IV.

If the system we are going to design works perfectly, interference of type 1 should not pose too big of a problem. Ideally we would estimate the interference during the training phase and then deal with it during data reception. Interference of type 2 should do even less harm: we have estimated it and are thus prepared to face it, but finally it is not even present during data reception. Interference of type 3 however is more difficult to tackle. It is not present during the training phase and we have thus no means to gather any knowledge about it whatsoever. We will thus need some additional mechanism to take care of type 3 interference. We will address estimation and mitigation of interference types 1 and 2 in subsections III-C and III-D. A possible solution to mitigate interference of type 3 is presented in subsection III-D.

B. Interference Models

As already presented in section I we consider two ways of increasing complexity to model interference: the Gaussian mixture model (GMM) and the hidden Markov model (HMM).

In the case of the GMM we assume that the interference and noise samples z_n are i.i.d random variables and that the vector $\mathbf{z} = (z_1, \ldots, z_N)$ has underlying probability distribution

$$f(\mathbf{z}|\mathbf{\Theta_z}) = \prod_{n=1}^{N} \sum_{p=1}^{P} \lambda_p \cdot \phi(z_n|\sigma_p^2) \tag{5}$$

where $\mathbf{\Theta_z}$ is the vector of model parameters, $\mathbf{\Theta_z} = (\mathbf{\Lambda}, \mathbf{\Sigma}) = (\lambda_1, \ldots, \lambda_P, \sigma_1^2, \ldots, \sigma_P^2)$, P is the model order specifying the number of mixture components, λ_p is the prior probability of component p and $\phi(z_n|\sigma_p^2)$ is the pth component density assumed to be zero-mean Gaussian with variance σ_p^2, i.e. $\phi(z_n|\sigma_p^2) = \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp(-z_n^2/2\sigma_p^2)$. The choice of a Gaussian with zero mean is motivated by the fact the we are considering BPSK modulation. However even with a modulation of non-zero mean, a random phase of the channel coefficients leads to samples of zero-mean.

The HMM introduces correlation, the samples z_n are no longer required to be independent. In addition to the sample vector $\mathbf{z}$, we now also have a hidden vector of states, $\mathbf{q} = (q_0, \ldots, q_N)$, with $q_n \in \{1, \ldots, P\}$. To each state q_n is associated a Gaussian component density $\phi(z_n|\sigma_{q_n}^2)$ defined as in the GMM case. Each state q_n determines which of the P component densities generates the sample z_n. The initial state is q_0. It is described by the vector of initial state probabilities $\mathbf{\Pi}$ with entries of the form $\pi_i = p(q_0 = i), i \in \{1, \ldots, P\}$. Transitions among the states occur according to a matrix of transition probabilities $\mathbf{A}$. An entry $a_{ij} = p(q_n = j|q_{n-1} = i)$ of $\mathbf{A}$ with $i, j \in \{1, \ldots, P\}$ is the probability that sample z_n is generated by component density j, knowing that z_{n-1} was generated by component density i. The vector $\mathbf{z}$ then has the following probability distribution

$$f(\mathbf{z}|\mathbf{\Theta_z}) = \sum_{\mathbf{q} \in \mathcal{Q}} \pi_{q_0} \prod_{n=1}^{N} a_{q_{n-1}q_n} \phi(z_n|\sigma_{q_n}^2) \tag{6}$$

where $\mathcal{Q}$ is the space of all possible state vectors and the vector of model parameters is now $\mathbf{\Theta_z} = (\mathbf{\Pi}, \mathbf{A}, \mathbf{\Sigma})$. Note that the GMM is only a special case of the HMM with $\mathbf{\Pi}$ arbitrary and $a_{ij} = \lambda_j$.

C. Training phase

Here we show how to estimate the statistics of interference that is present during the training phase (interference of types 1 and 2). In addition, we also estimate the channel. We thus want to find the maximum-likelihood estimate of $\mathbf{\Theta} = (\mathbf{\Theta_z}, \mathbf{\Theta_c})$ from the training sequence, where $\mathbf{\Theta_z}$ stands for the parameters of the Interference model and $\mathbf{\Theta_c} = (\beta_1, \ldots, \beta_M)$ the parameters of the channel[3]. The discrete time received signal is given by (4) and we find the sequence z_n during the training phase as

$$z_n = y_n - \sum_{k=1}^{N_t} d_k \sum_{m=0}^{M-1} \beta_m \cdot \delta(n - c_k - k(N_c + N_g) - m) \tag{7}$$

with $n = 1, \ldots, N$ and $N = N_t \cdot (N_c + N_g)$ the number of samples during the training phase. Note that z_n depends on the channel parameters. However, to ease notation, we will write z_n instead of $z_n(\mathbf{\Theta_c})$ whenever possible. We can now formulate the maximum-likelihood estimation problem as

$$\hat{\mathbf{\Theta}} = \arg\max_{\mathbf{\Theta}} \ln(f(\mathbf{z}|\mathbf{\Theta})) \tag{8}$$

where we chose to maximize the log-likelihood rather than the likelihood because it simplifies expressions and where f is replaced by (5) or (6) depending on the interference model. In general, direct maximization of (8) is difficult and the classical method of choice is the EM-algorithm [6]. The EM-algorithm is an iterative algorithm used to find the maximum-likelihood parameter estimate in situations where optimization of the likelihood function is simplified by assuming the existence of hidden data $\mathbf{x}$ in addition to the observation $\mathbf{z}$. The complete-data log-likelihood is then defined as $\ln(f(\mathbf{z}, \mathbf{x}|\mathbf{\Theta}))$. The EM-algorithm loops over the following steps:

1) E-Step: calculate the conditional expectation of the complete-data log-likelihood with respect to the hidden data $\mathbf{x}$ given the observed data $\mathbf{z}$ and the current parameter estimate $\hat{\mathbf{\Theta}}'$

$$Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}') = E\big[\ln(f(\mathbf{z}, \mathbf{x}|\mathbf{\Theta}))\big|\mathbf{z}, \hat{\mathbf{\Theta}}'\big] \tag{9}$$

2) M-Step: find the new parameter estimate as

$$\hat{\mathbf{\Theta}} = \arg\max_{\mathbf{\Theta}} Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}') \tag{10}$$

The parameter estimate found with the EM-algorithm is the solution of (8). Parameter Estimation for GMM and HMM by EM is a well-known and widely used procedure (see e.g. [7]). We now give the resulting algorithms for the two models under consideration.

1) EM-algorithm for the GMM: The hidden data sequence is $\mathbf{x} = (x_1, \ldots, x_N)$ where $x_n \in \{1, \ldots, P\}$ indicates which component density generated sample z_n. The random variables x_n are thus i.i.d. with $p(x_n = p) = \lambda_p$. Following similar procedures as the ones described in [5], [7], the algorithm looping over the following steps results

1) E-Step 1: calculate $\gamma_p(n) = P(x_n = p|z_n, \hat{\mathbf{\Theta}}'_{\mathbf{z}}, \hat{\mathbf{\Theta}}'_{\mathbf{c}})$
2) M-Step 1: find the new parameter estimate $\hat{\mathbf{\Theta}}_{\mathbf{z}}$ as

[3]In practice, the number of channel parameters can be significant. We will therefore not be able to estimate all of them, and accept to only estimate the first few ones.

$$\hat{\mathbf{\Theta}}_{\mathbf{z}} = \arg\max_{\mathbf{\Theta}_{\mathbf{z}}} Q((\mathbf{\Theta}_{\mathbf{z}}, \hat{\mathbf{\Theta}}'_{\mathbf{c}}), (\hat{\mathbf{\Theta}}'_{\mathbf{z}}, \hat{\mathbf{\Theta}}'_{\mathbf{c}}))$$

resulting in

$$\hat{\lambda}_p = \frac{1}{N}\sum_{n=1}^{N}\gamma_p(n), \quad \hat{\sigma}_p^2 = \frac{z_n(\hat{\mathbf{\Theta}}_{\mathbf{c}})^2 \cdot \gamma_p(n)}{\sum_{n=1}^{N}\gamma_p(n)} \tag{11}$$

3) E-Step 2: calculate $\gamma_p(n) = P(x_n = p|z_n, \hat{\mathbf{\Theta}}_{\mathbf{z}}, \hat{\mathbf{\Theta}}'_{\mathbf{c}})$
4) M-Step 2: compute $\hat{\beta}_0, \ldots, \hat{\beta}_M$ sequentially by

$$\hat{\beta}_0 = \arg\max_{\beta_0} Q((\hat{\mathbf{\Theta}}_{\mathbf{z}}, (\beta_0, \hat{\beta}'_1, \ldots, \hat{\beta}'_M)), (\hat{\mathbf{\Theta}}_{\mathbf{z}}, \hat{\mathbf{\Theta}}'_{\mathbf{c}}))$$

$$\hat{\beta}_1 = \arg\max_{\beta_1} Q((\hat{\mathbf{\Theta}}_{\mathbf{z}}, (\hat{\beta}_0, \beta_1, \ldots, \hat{\beta}'_M)), (\hat{\mathbf{\Theta}}_{\mathbf{z}}, \hat{\mathbf{\Theta}}'_{\mathbf{c}}))$$

and so on, resulting in

$$\hat{\beta}_m = \frac{\sum_{n=1}^{N} u_{n,m} v_n \sum_{p=1}^{P} \frac{\gamma_p(n)}{\hat{\sigma}_p^2}}{\sum_{n=1}^{N} v_n^2 \sum_{p=1}^{P} \frac{\gamma_p(n)}{\hat{\sigma}_p^2}} \tag{12}$$

with
$u_{n,m} = y_n - \sum_{k=1}^{N_t} d_k \sum_{\substack{\tilde{m}=0 \\ \tilde{m}\neq m}}^{M-1} \hat{\beta}_{\tilde{m}} \cdot \delta(n - c_k - k(N_c+N_g) - \tilde{m})$,
$v_n = \sum_{k=1}^{N_d} d_k \cdot \delta(n - c_k - k(N_c + N_g) - m)$ and where

$$\gamma_p(n) = P(x_n = p|z_n, \hat{\mathbf{\Theta}}') = \frac{\lambda'_p \cdot \phi(z_n(\hat{\mathbf{\Theta}}'_{\mathbf{c}})|\sigma_p^{2\prime})}{\sum_{\tilde{p}=0}^{P} \lambda'_{\tilde{p}} \cdot \phi(z_n(\hat{\mathbf{\Theta}}'_{\mathbf{c}})|\sigma_{\tilde{p}}^{2\prime})} \tag{13}$$

can be interpreted as the posterior probability that the interference and noise sample z_n was drawn from the Gaussian mixture with variance σ_p^2. Note that this is a modified version of the EM-algorithm that alternates between updating the interference model parameters $\hat{\mathbf{\Theta}}_{\mathbf{z}}$ and updating the channel parameters $\hat{\mathbf{\Theta}}_{\mathbf{c}}$. This simplifies the joint maximization of the parameters and is known as the space-alternating generalized EM-algorithm (SAGE) [8].

2) EM-algorithm for the HMM: In case of the HMM the hidden data sequence $\mathbf{x} = (x_0, \ldots, x_N) = \mathbf{q}$ where $\mathbf{q}$ is the hidden sequence of states. The random variables $x_n \in \{1, \ldots, P\}$ thus form a homogeneous Markov chain with initial state probabilities $\mathbf{\Pi}$ and transition matrix $\mathbf{A}$. The resulting algorithm turns out to have the same structure as in the case of the GMM. For a derivation of the update equations, the reader is referred to [7]. The update equations are

$$\hat{\pi}_p = \gamma_p(0), \quad \hat{a}_{ij} = \frac{\sum_{n=1}^{N}\xi_{i,j}(n)}{\sum_{n=1}^{N}\gamma_p(n)} \tag{14}$$

$\hat{\sigma}_p^2$ and $\hat{\beta}_m$ are found to be given again by (11) and (12). As opposed to the GMM we now have to calculate two quantities, $\gamma_p(n) = P(x_n = p|z_n, \hat{\mathbf{\Theta}}')$ and $\xi_{i,j}(n) = P(x_{n-1} = i, x_n = j|z_n, \hat{\mathbf{\Theta}}')$, during the E-step. Further there exists no closed form expression for the above quantities. However they can be determined via an iterative method known as the forward-backward or Baum-Welch algorithm [9].

D. Data Reception Phase

After having estimated the parameters of the interference model and the channel parameters, we can now use these estimates to recover the data sequence. In the following we are first going to show how interference of types 1 and 2 can be mitigated. This is done by using a decoder that takes advantage of the estimated interference model. We will then explain why this interference mitigation procedure is not effective against interference of type 3 and propose a possible solution. At the end of this subsection we will finally give the complete algorithm for the data reception phase.

1) Decoding using statistical interference modeling: Similar to the training phase, the sequence of the noise and interference terms during data reception is given by

$$z_n = y_n - \sum_{k=N_t+1}^{N_d} d_k \sum_{m=0}^{M-1} \beta_m \cdot \delta(n - c_k - k(N_c+N_g) - m) \tag{15}$$

with $n = 1, \ldots, N$ where $N = N_p \cdot (N_c + N_g)$ is the number of samples during the data reception phase. Assuming that the information symbols are equiprobable, the optimum decoding rule to recover the data sequence $\mathbf{d} = (d_{N_t+1}, \ldots, d_{N_d})$ is the maximum likelihood criterion. We thus want to find the data sequence $\hat{\mathbf{d}} = (\hat{d}_{N_t+1}, \ldots, \hat{d}_{N_d})$ that maximizes the likelihood of observing the sequence $\mathbf{z} = (z_1, \ldots, z_N)$

$$\hat{\mathbf{d}} = \arg\max_{\mathbf{d}} \ln(f(\mathbf{z}|\mathbf{d})) \tag{16}$$

Since the distribution of $\mathbf{z}$ is given by (5) or (6), depending on the model, this places us again in the framework of the EM-algorithm, which has been shown in [10] for a general context. Combining (9) and (15) and dropping all the terms not depending on $\mathbf{d}$ we obtain for both models

$$\tilde{Q}(\mathbf{d}, \hat{\mathbf{d}}') = \sum_{k=N_t+1}^{N_d}\sum_{m=0}^{M-1}\sum_{p=1}^{P} \beta_m \cdot y_{t_{k,m}} \cdot d_k \cdot \frac{\gamma_p(t_{k,m})}{\sigma_p^2} \tag{17}$$

where $\gamma_p(n) = p(x_n = p|z_n, \hat{\mathbf{d}}')$ and $t_{k,m} = c_k + k(N_c + N_g) + m$. Note that for the GMM the E-step remains exactly the same as in the training phase, only that the parameter to estimate is now $\mathbf{d}$ instead of $\mathbf{\Theta}$. For the case of the HMM the E-step is also similar to the training phase but we no longer have to calculate $\xi_{i,j}(n)$. The M-step is the same for both models, but different from the training phase as we are now maximizing (17) with respect to $\mathbf{d}$. We see from (17) that the maximization reduces to a classical max-sum problem, that can be solved by the Viterbi algorithm [11] with branch metric

$$m(d_k) = \sum_{m=0}^{M-1}\sum_{p=1}^{P} \beta_m \cdot y_{t_{k,m}} \cdot d_k \cdot \frac{\gamma_p(t_{k,m})}{\sigma_p^2} \tag{18}$$

Note that the receiver discussed here implicitly has the structure of a RAKE receiver performing maximum-ratio combining. This can be seen from (18): All of the estimated legs of the multipath channel contribute to the detection of the symbol. Further they are weighted according to their path gain, β_m, and an additional factor, $\gamma_p(t_{k,m})/\sigma_p^2$, accounting for interference. From this observation we can get a good intuition on how the algorithm actually mitigates interference. Recall that $\gamma_p(t_{k,m})$ is the posterior probability that the sample $y_{t_{k,m}}$ has an interference and noise term drawn from a zero-mean Gaussian with variance σ_p^2. In the above metric, samples with noise terms that stem with high probability from a distribution with high variance consequently get penalized through the

factor $\gamma_p(t_{k,m})/\sigma_p^2$. This factor plays the role of a weighting function: Samples with low interference level get a larger weight than samples that are likely to be polluted by a high interference and noise term. Note that this also applies to the estimation of the channel coefficients during the training phase. The same weighting factor appears in (12). Therefore we will also have a more reliable channel estimate when interference mitigation is performed, which has been shown in [5] for the case of the GMM. Interference of type 1 thus gets mitigated through the weighting function in (18) as well as indirectly through the better channel estimate. When facing interference of type 2, the weighting factor has less impact. In this case we mostly benefit from the better channel estimate.

2) Thresholding to mitigate interference type 3: As already mentioned, the situation is different for interference of type 3. It is not present during training and therefore the estimated variances of the interference and noise term will be rather small (in the order of the background noise variance). Samples with a lot of interference will thus still get a relatively high weight. This observation is the key to our solution to mitigate interference of type 3. After the training phase we determine the largest of the estimated variances. Assume this is σ_{p*}^2. We then determine a threshold ν, such that $P(X \geq \nu) \leq \varepsilon$, where $X \sim \mathcal{N}(0, \sigma_{p*}^2)$ and ε is some predetermined small probability. After each E-step we set

$$\gamma_p(n) = 0; \quad \forall p, n \text{ such that } z_n > \nu$$

This ensures that samples, that with high probability cannot be explained by the estimated interference model, do not contribute to the branch metric. As this is likely to affect predominantly samples polluted by interference of type 3, we have found a way to mitigate this type of interference.

To summarize 1) and 2), this results in the following algorithm for the data reception phase:

1) Initialization: Determine the threshold ν. Take an initial guess, $\hat{\mathbf{d}}^{(\mathbf{0})}$, for $\mathbf{d}$
2) E-Step: calculate $\gamma_p(n)$ in the same way as in the training phase (i.e. using (13) in the case of the GMM, using the forward-backward algorithm in the case of the HMM).
3) Thresholding: set $\gamma_p(n)$ to zero for any sample z_n that lies above the threshold ν.
4) M-Step: find the new parameter estimate $\hat{\mathbf{d}}$ by means of the Viterbi algorithm with metric given by (18).
5) Repeat steps 2 to 4 until convergence

IV. Performance Evaluation

In this section we evaluate the performance of the proposed receiver through simulations. The performance metric is the bit error rate (BER) versus signal to noise ratio (SNR, defined as $\frac{E_b}{N_0}$) at the receiver. Our simulation setup is the following. $T_c = 2$ns, $N_c = N_g = 128$, which results in a pulse repetition frequency of 1.95MHz. Channel coding is performed with simple pulse repetition codes of rates $1/2$ and $1/4$, resulting in physical layer peak data rates of 0.98Mb/s and 0.49Mb/s, respectively. The pulse shape $p(t)$ is chosen to be the second derivative of a Gaussian monopulse given by $p(t) = (1 - 4\pi(t^2/\tau^2)) \cdot \exp(-2\pi t^2/\tau^2)$ with $\tau = 0.7$. The channel model

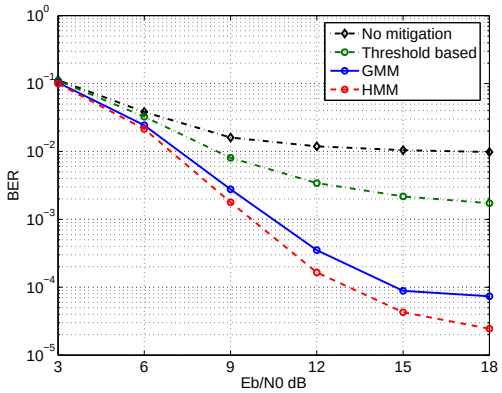

Fig. 1. Comparison of different receivers. 'No mitigation': a receiver that does not mitigate multi-user interference (MUI) but makes a Gaussian approximation. 'Threshold based': a receiver performing mitigation without modeling but only with simple thresholding. 'GMM' and 'HMM': our interference mitigation technique that employs a combination of thresholding and interference modeling using a Gaussian Mixture Model and a Hidden Markov Model, respectively. Results are for a pulse repetition code with rate $1/4$. Physical layer packets are generated at half the peak data rate. It can be seen, that the performance gain from modeling the interference is significant. Using the more sophisticated HMM to model MUI gives an additional gain compared to the GMM model.

is the Saleh-Valenzuela model proposed for 802.15.4a [12]. We ran simulations for the indoor office LOS and NLOS scenarios but as we did not observe fundamental differences between these two channel models we only show the results for the NLOS model. Physical layer data packets are assumed to be generated by a homogeneous Poisson process with rate λ, where λ either corresponds to the full physical layer peak data rate, half or one fourth of the peak data rate. Each packet has a length of 127 bytes or $N_d = 1016$ data symbols. The first $N_s = 112$ symbols are assumed to be reserved for the synchronization preamble, the next $N_t = 32$ symbols form the training sequence and the remaining $N_p = 872$ symbols constitute the payload. Two successive packets are assumed to be at least separated by the duration of a packet of size $N_s + N_t$, leaving room for acknowledgements. To simulate interference, a near-far scenario with $N_u = 5$ users was chosen. At the receiver, the four interferers have power levels of -3dB, 0dB, 10dB and 20dB with respect to the user of interest. The model order of the interference models is fixed to $P = 2$. We also ran simulations with a model order of $P = 3$. This did not change the results fundamentally, we observed a slight improvement for the HMM and no improvement for the GMM, and therefore we do not show these results here. Initialization of the estimators in the training phase is done as in [5], i.e. $\hat{\lambda}^{(\mathbf{0})} = (0.99, 0.01)$, $\hat{\sigma}_1^{(0)} = \frac{1}{N}\sum_{n=1}^{N} z_n^2$, $\hat{\sigma}_2^{(0)} = 50 \cdot \hat{\sigma}_1^{(0)}$ and $\hat{\beta}_m^{(0)} = \frac{1}{\sqrt{M}}$. The estimator for the data sequence $\hat{\mathbf{d}}$ is initially simply set to zero. The probability governing the threshold is set to $\varepsilon = 10^{-4}$ and the number of estimated channel parameters to $M = 28$, values which were both found through simulations.

To compare our solution with existing ones we also simulate a receiver that does not mitigate interference and thus makes a Gaussian approximation. This is equivalent to a receiver with a model order of $P = 1$ and without thresholding. To simulate a receiver that only employs simple thresholding without modeling interference, we again simulate a receiver with one

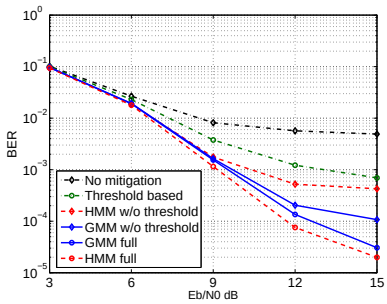

Fig. 2. Here we show the effect of interference type 3. We use a repetition code of rate $1/4$ and assume packet generation at one fourth of the peak rate. We see that whether or not to perform thresholding in addition to modeling interference greatly affects performance. Especially for the HMM the effect is huge: Without thresholding it barely performs better than the simple receiver performing only thresholding.

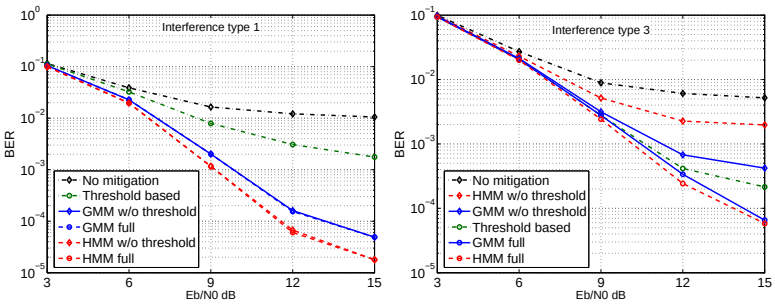

Fig. 3. Here we have categorized all packets depending on the type of interference they experience and show the results for types 1 and 3 separately. The simulation setup is the same as in Figure 2. The plots confirm our hypothesis that the losses seen in Figure 2 when not using thresholding are almost exclusively due to packets that face interference of type 3. Note that for interference type 3 both GMM and HMM without thresholding, perform worse than the simple threshold based receiver, which does not model interference.

single state. However, this time we also perform thresholding as explained in section III-D.2 and set the threshold based on the estimated variance of the single state.

Our simulations show that a 2-state model achieves a significant performance gain over traditional techniques that do not account for MUI or use a simple threshold mechanism. In Fig. 1 we show the BER for code rate $1/4$ and λ equal to half the peak data rate. We see that the proposed receiver obtains a significant gain, even at low SNRs. The difference between the GMM and the HMM however is much less pronounced. Still, the HMM gets a performance gain indicating that it is better suited to model MUI.

Further, we find that (1) modeling of interference alone is not sufficient, thresholding is still needed to prevent losses, and (2) these losses are effectively due to type 3 interference, it is thus important to make a classification of the interference and mitigate each type accordingly. Fig. 2 shows results for a code rate of $1/4$ and a packet rate, λ, of $1/4$ of the peak rate. Due to the less frequent packet transmissions, generally less interference occurs. This makes the difference between the schemes smaller. We further show the curves for the GMM and the HMM when thresholding is omitted. We have a clear performance degradation in both cases. The HMM even performs worse than the GMM and only slightly better than the threshold based receiver. We interpret this to be due to the HMM modeling MUI better and therefore being less flexible, when encountering situations where the predicted model not applies. Fig. 2 thus shows that the thresholding mechanism is needed even when modeling interference. To confirm that the performance degradation in Fig. 2 effectively is due to interference of type 3, we classified all data packets with respect to the interference type they were facing. This allows us to plot the BER curve for each of the types separately, which is shown in Fig. 3. We see that for packets with type 1 interference, the threshold has no impact. Packets with type 3 interference however suffer from a large loss if the threshold is not applied. Note that in this case both the HMM and the GMM perform worse than the threshold based receiver. This further confirms that better modeling makes the receiver less flexible in situations where the model no longer applies, which can be overcome by using the proposed thresholding algorithm.

V. Conclusion

This paper being more of theoretical nature, there are several aspects that have been omitted on purpose. First of all we did not do a complexity analysis for the choice of the interference model. The HMM requires much more processing than the GMM due to the forward-backward algorithm in the E-step. Also from a viewpoint of energy consumption, the GMM has an advantage. During chips where the signal of interest is not present the receiver can be turned off. The interference level can later still be estimated as there is no time-dependence between the samples. There are also some aspects we leave for future work. It might be interesting to investigate a non-data-aided approach where the estimation is performed constantly on-the-fly. Also one could imagine estimating the interference during idle periods. Both methods could be interesting to further reduce interference of type 3.

References

[1] R. Merz, J. Widmer, J.-Y. Le Boudec, and B. Radunovic, "A Joint PHY/MAC Architecture for Low-Radiated Power TH-UWB Wireless Ad-Hoc Networks," *WCMC Journal*, 2005.

[2] B. Hu and N. Beaulieu, "Accurate evaluation of multiple-access performance in TH-PPM and TH-BPSK UWB systems," *IEEE Trans. Commun.*, vol. 52, no. 10, pp. 1758–1766, Oct. 2004.

[3] G. Durisi and G. Romano, "On the validity of gaussian approximation to characterize the multiuser capacity of UWB TH PPM," in *2002 IEEE Conference on Ultra Wideband Systems and Technologies*, pp. 157– 161.

[4] R. Blum, R. Kozick, and Sadler, "An adaptive spatial diversity receiver for non-Gaussian interference and noise," *IEEE Trans. Signal Processing*, vol. 47, no. 8, pp. 2100 – 2111, Aug. 1999.

[5] V. Cellini and G. Dona, "A novel joint channel and multi-user interference statistics estimator for UWB-IR based on Gaussian mixture model," in *ICU 2005*, Sept. 5–8, 2005, pp. 655–660.

[6] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[7] J. Bilmes, "A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," 1997. [Online]. Available: citeseer.ist.psu.edu/bilmes98gentle.html

[8] J. Fessler and A. Hero, "Space-alternating generalized expectation-maximization algorithm," *IEEE Trans. Signal Processing*, vol. 42, no. 10, pp. 2664–2667, Oct. 1994.

[9] L. E. Baum, T. Petrie, G. Souled, and N. Weiss, "A maximization technique occurring in statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Stat.*, vol. 41, no. 1, pp. 164–171, 1970.

[10] W. Turin, "MAP decoding in channels with memory," *IEEE Trans. Commun.*, vol. 48, no. 5, pp. 757–763, May 2000.

[11] G.-D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, 1973.

[12] A. F. Molisch *et al.*, "IEEE 802.15.4a channel model - final report," Nov. 2004. [Online]. Available: ieee802.org/15/pub/TG4a.html